# Computationally-Sound Proofs

Silvio Micali

Department of Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
silvio@theory.lcs.mit.edu

**Abstract.** This paper puts forward a new notion of a proof based on computational complexity, and explores its implications to computation at large.

Computationally-sound proofs provide, in a novel and meaningful framework, answers to old and new questions in complexity theory. In particular, given a random oracle or a new complexity assumption, they allow us to prove that verifying is easier than deciding; to provide a quite effective way to prove membership in computationally hard languages (such as $\mathcal{C}o\text{-}\mathcal{NP}$-complete ones); and to show that every computation possesses a short certificate vouching its correctness.

## 1 Introduction

**A new notion.** Proofs are fundamental to our lives, and as for all things fundamental we should expect that answering the question of what a proof is will always be an on-going process. Indeed, we wish to put forward the new notion of a *computationally-sound proof* (*CS proof* for brevity) which achieves new and important goals, not attained or even addressed by previous notions.

Informally, a CS proof of a statement $S$ consists of a short string $\sigma$, very easy to verify and as easy to find as possible, offering a strong computational guarantee about the verity of $S$. By "very easy to verify" we mean that the time necessary to inspect a CS Proof of a statement $S$ is poly-logarithmically shorter than that required to decide $S$. By "as easy to find as possible" we mean that a CS proof of a *true* statement (i.e., for the purposes of this paper, *derivable* in a given axiomatic theory) can be computed in a time essentially comparable to that needed to decide the statement. Finally, by saying that the guarantee offered by a CS proof is "computational" we mean that false statements either do not have any CS proofs, or such "proofs" are practically impossible to find.

**Implementations of CS proofs.** The value of a new notion, of course, crucially depends on whether it can be sufficiently exemplified. We provide two